

American International University-Bangladesh (AIUB)

**Department of Computer Science  
Faculty of Science & Technology (FST)**

**Automated Irrigation System**

A Software Quality and Testing Project Submitted

By

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Semester: Fall\_23\_24** | | | | **Section: C** | **Group No: 07** |
| **SL** | **SN** | **Student Name** | **Student ID** | Individual  Contribution (in %) | Total Marks: 50 |
| Earned Marks: |
| **A** | 22 | AMIN, MD. AMINUL ISLAM | 21-45230-2 | 25% |  |
| **B** | 23 | BORNO, MEHERAB HASSAN | 21-45236-2 | 25% |  |
| **C** | 24 | HOSSAIN, MD. NAJIB | |  | | --- | |  |   21-45366-2 | 25% |  |
| **D** | 29 | KHAN, MD. SHADMAN TAHSIN | 21-45796-3 | 25% |  |

The project will be Evaluated for the following Course Outcomes

|  |  |
| --- | --- |
| **EVALUATION CRITERIA** | **Total Marks (50)** |
|  |
| Application Features | [10 Marks]  **A: B: C: D:** |
| Quality Attributes | [10 Marks]  **A: B: C: D:** |
| Plan for Testing Levels | [10 Marks]  **A: B: C: D:** |
| Test cases, Format, Submission | [20 Marks]  **A: B: C: D:** |

Software Test Plan

for

Automated Irrigation System

Version 1.0 approved

Prepared by MD. AMINUL ISLAM AMIN, MEHERAB HASSAN BORNO, MD. NAJIB HOSSAIN, MD. SHADMAN TAHSIN KHAN.

American International University-Bangladesh (AIUB)

September 21, 2024.

Table of Contents

[1. Application Features 3](#_Toc177553070)

[2. Application Quality Attributes 4](#_Toc177553071)

[3. Testing Levels 4](#_Toc177553072)

[4. TEST CASES/TEST ITEMS 4](#_Toc177553073)

# Application Features

**1. Registration**

**Functional Requirements:**

1.1 The application shall allow new users to register by providing a valid email, username, and password.  
1.2 The application shall validate the entered email for correct format and check for existing accounts.  
1.3 The password must meet the application’s security requirements (e.g., minimum 8 characters, etc.).  
1.4 Upon successful registration, the application shall send a confirmation email to the user's email address. 1.5 After confirming the email, the user should be able to log into the application.

Priority Level: Medium  
Precondition: User must have access to a valid email address.

Cross reference: 3.1, 3.3, 4

**2. Login**

**Functional Requirements:**

2.1 The Application shall allow users to log in using a valid username and password.  
2.2 If the entered password is incorrect, the Application shall notify the user and allow up to 3 retries.  
2.3 Upon successful login, the Application shall establish a session and redirect the user to the dashboard.

2.4 If login fails after 3 times attempts, the user account shall be locked for 1 hour.

Priority Level: Medium  
Precondition: User must have a valid registered username and password.

Cross reference: 3.1, 3.2, 3.3, 4

**3. Weather conditions Integration**

**Functional Requirements:**

3.1 The Application shall display real-time weather conditions, including temperature, humidity, wind speed, and rain probability.  
3.2 Users can change their preferred location to view conditions for different cities.

Priority Level: Low  
Precondition: The Application should be integrated with a weather API service.

Cross reference: 3.2, 3.3, 4

**4. Settings**

**Functional Requirements:**

4.1 The Application shall allow users to set their preferred location for weather updates and other relevant notifications.  
4.2 The Application shall allow users to view and update their profile information, such as name and email address.  
4.3 The Application shall allow users to change their password.

Priority Level: Medium  
Precondition: The user must be logged in to access settings.

Cross reference: 3.3, 4

**5. Community Feedback**

**Functional Requirements:**

5.1 Users shall be able to rate the app and leave reviews.  
5.2 Developers can respond to user feedback within the app, and users will be notified of responses.

Priority Level: Low  
Precondition: Users must have an active account.

Cross reference: 4

**6. Soil Moisture Monitoring**

**Functional Requirements:**

6.1 The Application will allow users to check the moisture levels in their soil in real-time.  
6.2 The Application shall display historical data of soil moisture levels for analysis.

Priority Level: High  
Precondition: Soil moisture sensors should be properly integrated and functional.

Cross reference: 3.1, 3.3, 3.4, 4

**7. Notifications**

**Functional Requirements:**

7.1 The Application shall send notifications for important events, such as weather alerts or reminders to check soil moisture levels. 7.2 The Application shall send notifications when there is a response from the helpline or support center.

Priority Level: Medium  
Precondition: Users must have an active account to receive notifications.

Cross reference: 3.2, 3.3, 3.4, 4

**8. Helpline Support**

**Functional Requirements:**

8.1 The Application shall provide access to a helpline for user assistance.  
8.2 Users can contact support via email or phone for queries or issues.

Priority Level: Medium  
Precondition: Users must have an active account to access personalized support.

Cross reference: 4

**9. Irrigation Application Control**

**Functional Requirements:**

9.1 The Application shall be able to turn the irrigation system on and off automatically, based on pre-defined settings such as weather conditions or soil moisture levels.  
9.2 Users shall be able to manually control the irrigation Application (turn on/off).  
9.3 The Application shall allow users to set a timer for irrigation start and stop times.  
9.4 Users shall be able to set a preferred water level for irrigation using this Application.

Priority Level: High  
Precondition: The irrigation Application must be connected to the control module.

Cross reference: 3.1, 3.2, 3.3, 3.4, 4

**10. Logout**

**Functional Requirements:**

10.1 The Application shall allow users to log out at any time from the dashboard or other pages.  
10.2 Upon logout, the Application shall terminate the active session and redirect the user to the login page.  
10.3 After logout, the Application shall invalidate any active session tokens to prevent unauthorized access.

Priority Level: Medium  
Precondition: User must be logged in.

Cross reference: 3.4, 4

# Application Quality Attributes

#### QA1 - ****Usability****

A trained user shall be able to view and control the irrigation Application, monitor weather conditions, and access soil moisture data within an average of 5 minutes and a maximum of 8 minutes. The interface shall be accessible for users with minimal agricultural technology experience.

#### QA2 - ****Performance Efficiency****

The Application shall handle up to 500 current users accessing real-time weather and soil moisture data without a significant drop in performance. Response time for critical actions (e.g., starting or stopping irrigation) shall be within 5 seconds, and for data retrieval (e.g., weather conditions, soil moisture history), it shall be within 10 seconds.

#### QA3 - ****Reliability****

The Application shall have an uptime of 99%, ensuring that users can access weather data, soil moisture readings, and control irrigation with minimal interruptions. In case of network failures, critical features (such as automated irrigation based on predefined conditions) shall function offline using locally stored settings.

#### QA4 - ****Security****

The Application shall ensure any unauthorized login attempt by sending verification code to authorized user’s email. The Application shall ensure secure login using industry-standard encryption methods for password storage and transmission. It shall prevent unauthorized access to the control of the irrigation Application by enforcing strong password policies for critical operations like manual irrigation control.

#### QA5 - ****Scalability****

The Application shall be scalable to support future growth in both the number of users and the integration of additional sensors (e.g., multiple soil moisture sensors per user). The architecture shall be capable of supporting up to 10,000 users and a 200% increase in data points without significant performance degradation.

#### QA6 - ****Availability****

The Application shall be available to users 24/7, with a maximum downtime of 5 hours per month for maintenance or unexpected Application failures. Notifications for planned maintenance shall be sent to users.

#### QA7 - ****Maintainability****

Developers shall be able to perform Application updates, such as adding new features or patching security vulnerabilities, without more than 2 hours of Application downtime. The codebase shall follow standard software development practices, making it easily maintainable and extendable.

#### QA8 - ****Portability****

The Application shall be accessible through multiple platforms, including desktop browsers (Chrome, Firefox, Edge, Opera) and mobile devices (iOS, Android). The user interface shall be completely responsive without compromising usability or functionality.

#### QA9 - ****Responsiveness****

The Application shall be capable of sending notifications (e.g., check soil moisture, weather alert) within 30 seconds of receiving sensor data or external weather updates, ensuring timely information delivery to users for critical decisions.

# Testing Levels

**3.1 Unit Testing**

**User Registration & Login:** Test that the user registration and login processes work properly, including validation for fields such as username, email, and password.

**Weather conditions:** Test that the weather data is retrieved and displayed accurately for the chosen location.

**Soil Moisture Monitoring:** Test that the soil moisture sensor data is processed and displayed correctly for different soil types.

**Irrigation Application Control:** Ensure that the automatic and manual control features for the irrigation Application work correctly.

**3.2 Integration Testing**

**User Login and Soil Monitoring Integration:** Ensure that once a user logs in, they can access the soil moisture data without issues.

**Weather conditions and Irrigation Control Integration:** Test that the irrigation Application can adjust based on real-time weather conditions (e.g., stopping the irrigation Application if it’s raining ).

**Notification Application:** Verify that the notification Application works properly across the app, sending alerts when the soil moisture is low, or any weather conditions are detected.

**3.3 System Testing**

**Login and Registration:** Ensure users can register, log in, and log out properly.

**Weather conditions Integration:** Test real-time weather data and location switching.

**Soil Moisture Monitoring:** Ensure accurate data capture from sensors and correct display in the application.

**Irrigation System Control:** Test the automated and manual control functionalities based on moisture levels and weather.

**Notifications:** Validate notification system (weather alerts, moisture checks).

**Settings:** Verify the functionality of changing location, personal information and old password.

**3.4 Acceptance Testing**

**User Interface and Experience:** Ensure the app is easy to navigate and that users can easily set up their soil moisture preferences, review weather conditions, and control irrigation.

**Functionality Verification:** Confirm that core functions like soil moisture monitoring, weather condition checking, and automatic or manual irrigation control work as expected.

**Notifications:** Verify that all relevant notifications (e.g., weather alerts, soil moisture check) are delivered on time.

**Real-World Scenarios:** Test the Application in real-world environments where it is expected to be used (e.g., fields or greenhouses) to ensure the application performs accurately in its operational environment.

# 4. TEST CASES/TEST ITEMS

**Table 1:** Test Case for Registration.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Automated Irrigation System | | | Test Designed by: Md.Shadman Tahsin Khan | | |
| Test Case ID: FR\_1 | | | Test Designed date: 20-sept-2024 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: Md.Aminul Islam Amin | | |
| Module Name: Registration | | | Test Execution date: 20-sept-2024 | | |
| Test Title: Verify registration with valid data. | | |  | | |
| Description: Test the user registration form by entering valid user data. | | |  | | |
| Precondition (If any): User should not already have an account. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to the registration page.  2. Fill in all required fields.  3. Click register button. | Name: Amin Islam  Username: Amin9  Email: amin9@gmail.com  Password: @123456A | User’s unique account should be created and now able to login. | | As expected, | Pass |
| Post Condition: The user should receive a confirmation code in email to verify the account. | | | | | |

**Table 2:** Test Case for Login.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Automated Irrigation System | | | Test Designed by: Md.Najib Hossain | | |
| Test Case ID: FR\_2 | | | Test Designed date: 20-sept-2024 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: Md.Shadman Tahsin Khan | | |
| Module Name: Login | | | Test Execution date: 20-sept-2024 | | |
| Test Title: Verify login with valid credentials. | | |  | | |
| Description: Test login with valid credentials. | | |  | | |
| Precondition (If any): User must be registered with valid credentials. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Go to the login page.  2. Enter a valid username and password.  3.Click login button. | Username: Amin9  Password: @123456A | User should be logged into the Application. | | As expected, | Pass |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database. | | | | | |

**Table 3:** Test Case for Weather conditions Integration.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Automated Irrigation System | | | Test Designed by: Meherab Hassan Borno | | |
| Test Case ID: FR\_3 | | | Test Designed date: 20-sept-2024 | | |
| Test Priority (Low, Medium, High): Low | | | Test Executed by: Md.Aminul Islam Amin | | |
| Module Name: Weather conditions Integration. | | | Test Execution date: 20-sept-2024 | | |
| Test Title: Weather conditions Integration. | | |  | | |
| Description: Verify weather conditions data retrieval. | | |  | | |
| Precondition (If any): User is validated with database and successfully login to account for weather condition check. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to the weather condition page.  2. Change the location. | Location: Khulna. | The current weather conditions should be displayed for preferred location. | | As expected, | Pass |
| Post Condition: Check if weather information is displayed correctly for the exact location. | | | | | |

**Table 4**: Test Case for Settings.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Automated Irrigation System | | | Test Designed by: Md.Shadman Tahsin Khan | | |
| Test Case ID: FR\_4 | | | Test Designed date: 20-sept-2024 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: Meherab Hassan Borno | | |
| Module Name: Settings | | | Test Execution date: 20-sept-2024 | | |
| Test Title: Verify that the user can update location, profile information, and current password. | | |  | | |
| Description: Test the functionality of the Settings module. | | |  | | |
| Precondition (If any): User must be logged in. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1.Go to the settings page.  2.Select the option to update location.  3.click on the update button.  4. Select the option to update profile info.  5. Update the user profile info (e.g., name, email).  6.click on the update button.  7. Select the option to change the old password.  8. Enter the old and new passwords.  9.click on the update button. | New location: Dhaka.  Name: Md. Aminul Islam  Email: maminul@gmail.com  old Password: @123456A  New Password: #Amin999 | The settings page is displayed.  Location selection option is displayed. System saves and updates the location. Location is successfully updated.  Profile information is editable.  System saves and updates the profile info.  Password change form is displayed.  System accepts the new password.  Password is successfully updated. | | As expected, | Pass |
| Post Condition: The Application should notify the changes immediately and must be persisted in the database for future sessions. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Automated Irrigation System | | | Test Designed by: Md.Najib Hossain | | |
| Test Case ID: FR\_5 | | | Test Designed date: 20-sept-2024 | | |
| Test Priority (Low, Medium, High): Low | | | Test Executed by: Md.Aminul Islam Amin | | |
| Module Name: Community Feedback. | | | Test Execution date: 20-sept-2024 | | |
| Test Title: Verify Community Feedback Submission. | | |  | | |
| Description: Test the ability to submit feedback and view developer responses. | | |  | | |
| Precondition (If any): Users must have an active account. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open the feedback section in the app.  2. Enter a rating and write a review.  3.Click on the Submit button.  4. Developer responds to the feedback.  5. Check the notification for developer response and open feedback section to view the response. | Rating: 5 stars  Review: "Great app!"  Developer Response: "Thank you!" | A feedback form opens, allowing the user to submit feedback.  Rating and review fields are filled and displayed correctly.  Feedback is submitted successfully, and confirmation is displayed.  Notification is sent to the user regarding the developer's response.  Developer’s response is visible in the feedback section under the user's review. | | As expected, | Pass |
| Post Condition: The application should notify the user when a developer responds to their feedback & should be able to view and acknowledge the developer’s response. | | | | | |

**Table 5:** Test Case for Community Feedback.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Automated Irrigation System | | | Test Designed by: Md.Aminul Islam Amin | | |
| Test Case ID: FR\_6 | | | Test Designed date: 20-sept-2024 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Md.Najib Hossain | | |
| Module Name: Soil Moisture Monitoring. | | | Test Execution date: 20-sept-2024 | | |
| Test Title: Verify Soil Moisture Monitoring Functionality. | | |  | | |
| Description: Test the soil moisture Monitoring functionality. | | |  | | |
| Precondition (If any): Soil moisture sensors should be properly integrated and functional. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open the soil moisture monitoring page.  2.Select the soil type from the options provided.  3.View the current soil moisture level for the selected soil type.  4.Check historical data for the past week to analyze soil moisture trends.  5.Check if the system provides a notification when moisture levels drop below a certain threshold. | Soil type: Sandy, Loamy, Clay.  Soil type: Loamy  Threshold set at 30% moisture. | The Soil moisture page opens successfully, showing real-time data display functionality.  The selected soil type is applied, and moisture readings are updated accordingly.  Real-time soil moisture level is displayed on the screen.  Moisture data for the past 7 days is displayed clearly for analysis.  System displays or sends a notification if the moisture level drops below the threshold. | | As expected, | Pass |
| Post Condition: Historical soil moisture data should be available for analysis & The application should send alerts or notifications to the user. | | | | | |

**Table 6:** Test Case for Soil Moisture Monitoring.

**Table 7:** Test Case for Notifications.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Automated Irrigation System | | | Test Designed by: Meherab Hassan Borno | | |
| Test Case ID: FR\_7 | | | Test Designed date: 20-sept-2024 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: Md.Shadman Tahsin Khan | | |
| Module Name: Notifications | | | Test Execution date: 20-sept-2024 | | |
| Test Title: Verify Notifications Functionality. | | |  | | |
| Description: Test the notifications system for weather alerts, soil moisture reminders, and helpline responses. | | |  | | |
| Precondition (If any): User is logged in, notifications are enabled. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Enable notifications for important events like weather alerts or soil moisture checks.  2. Check if the application sends a weather alert based on current data, soil moisture falls below the threshold, a response from the helpline/support center.  3.Verify the notification panel displays the messages for weather alerts and helpline responses. | Toggle notifications “ON”  Weather alert (e.g., rainfall), Soil moisture < 30%, Helpline response: "Your issue is resolved" | Notifications are enabled for the user.  User receives a weather alert notification on the device.  User receives a notification to check the soil moisture levels.  User receives a notification when the helpline responds to their inquiry.  Notifications are clearly visible in the notification panel of the app. | | As expected, | Pass |
| Post Condition: The application should store and display all notifications in the user’s notification panel for easy reference. | | | | | |

**Table 8:** Test Case for Helpline Support.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Automated Irrigation System | | | Test Designed by: Md.Shadman Tahsin Khan | | |
| Test Case ID: FR\_8 | | | Test Designed date: 20-sept-2024 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: Md.Najib Hossain | | |
| Module Name: Helpline Support. | | | Test Execution date: 20-sept-2024 | | |
| Test Title: Verify Helpline Support Functionality. | | |  | | |
| Description: Test the access to helpline services and support via email or phone. | | |  | | |
| Precondition (If any): Users must have an active account and be logged in to access the helpline. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Navigate to the helpline or support section in the app.  2. Select the email option to contact support and send a query.  3.Verify if a confirmation message appears after sending the email.  4. Select the phone option and check if the contact number is displayed for user assistance.  5. Verify if a notification is received when the helpline responds to an email query. | Email: maminul@gmail.com  Query sent.  Phone number: 01779922880  Response: "Issue resolved". | The helpline page is displayed with options for email or phone contact.  The email query is successfully sent to the support team.  A confirmation message "Your query has been sent to support" is displayed.  The helpline contact number is visible, allowing the user to call for assistance.  The user receives a notification indicating that the helpline has responded. | | As expected, | Pass |
| Post Condition: The application should store all user queries and responses for future reference, and the helpline support must be available during operational hours. | | | | | |

**Table 9:** Test Case for Irrigation Application Control.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Automated Irrigation System | | | Test Designed by: Meherab Hassan Borno | | |
| Test Case ID: FR\_9 | | | Test Designed date: 20-sept-2024 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Md.Shadman Tahsin Khan | | |
| Module Name: Irrigation Application Control. | | | Test Execution date: 20-sept-2024 | | |
| Test Title: Verify Irrigation System Control Functionality. | | |  | | |
| Description: Test the automatic and manual control of the irrigation system, timer settings, and water level adjustments. | | |  | | |
| Precondition (If any): The irrigation system must be connected to the control module, and the user must be logged in. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Navigate to the irrigation system control section.  2. Enable the “automatic irrigation control” option based on weather or soil moisture levels.  3. Manually turn“ON” the irrigation system.  4. Manually turn“OFF” the irrigation system.  5. Set a timer to start the irrigation at a specific time (e.g., 7:00 AM) and stop at 9:00 AM.  6. Set a preferred water level for the irrigation system.  7. Verify the system responds accurately to changes in pre-defined weather or soil conditions. | Weather: Rainy, Soil Moisture: Low.  Manual toggle: “ON”  Manual toggle: “OFF”  Start: 07:00 AM, Stop: 09:00 AM.  Water level: 60%.  Soil Moisture: High, Weather: Dry. | Irrigation control page is displayed with all control options.  Irrigation systems turn on/off automatically based on pre-defined conditions.  The irrigation system turns on, and the user receives confirmation.  The irrigation system turns off, and the user receives confirmation.  Irrigation system turns on/off as per the scheduled timer settings.  The irrigation system adjusts the water flow to the specified level (60%).  Irrigation turns off when soil moisture is high or turns on when dry weather. | | As expected, | Pass |
| Post Condition: The irrigation system should return to its default state if there is any disconnection or failure, and the user must be notified of any changes or interruptions. | | | | | |

**Table 10:** Test Case for Log out.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Automated Irrigation System | | | Test Designed by: Md.Aminul Islam Amin | | |
| Test Case ID: FR\_10 | | | Test Designed date: 20-sept-2024 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: Meherab Hassan Borno | | |
| Module Name: Log out. | | | Test Execution date: 20-sept-2024 | | |
| Test Title: Verify Logout functionality. | | |  | | |
| Description: Test the user logout functionality, session termination, and redirection to the login page. | | |  | | |
| Precondition (If any): The user must be logged in. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Navigate to the dashboard or any page with a logout option.  2. Click the logout button.  3. Attempt to access any page of the application after logout using the previous session.  4. Ensure that the session tokens are invalidated after logout to prevent re-access. |  | The logout button is visible and accessible.  User is logged out, the session is terminated, and user is redirected to the login page.  Access is denied, and the user is redirected to the login page.  Session tokens are invalidated, and unauthorized access is prevented. | | As expected, | Pass |
| Post Condition: The user should be successfully logged out, and any attempts to access restricted areas should prompt a login page. | | | | | |